SSLib: A Programming Environment for the Fitting and Analysis of Point Process and Hidden Markov Models

David Harte
Statistics Research Associates
Wellington
New Zealand

email: david@statsresearch.co.nz

Abstract

The Statistical Seismology Library (SSLib) is a collection of packages written in the R language. Two of the packages are PtProcess and HiddenMarkov, for the fitting and evaluation of point process and hidden Markov models, respectively. Both packages include some commonly used models, for example, the ETAS and stress release models, and also the Markov modulated Poisson process.

However, both packages are set up in a manner so that additional models can be added relatively easily to the existing framework. We will show how this works, and how we intend to develop this in the future.

Contributors

Mark Bebbington, Ray Brownrigg, Edwin Choi, Robert Davies, Michael Eglinton, David Harte, Dongfeng Li, Li Ma, Alistair Merrield, Andrew Tokeley, David Vere-Jones, Wenzheng Yang, Leon Young, Irina Zhdanova and Jiancang Zhuang

Introduction

SSLib is a collection of packages written in the R language

- R contains many standard routines (like IMSL or NAG)
- modular structure, components adhering to a common format
- code visible to user, can be added to and modified
- routines have consistent style of documented with examples
- provides some "documentation" of our analyses

Earthquake Catalogues

Catalogues have a standard format

Example: Sumatra (Nias) Earthquake of 28 March 2005:

```
library(sslib)
library(ssPDE)

usr <- c(92, 104, -5, 7)
a <- subset.rect(PDE, minlong=usr[1], maxlong=usr[2],
                 minlat=usr[3], maxlat=usr[4], minmag=4,
                 minday=julian(3,1,2005), maxday=julian(8,1,2005))

epicentres(a, usr=usr, magnitude=c(4,5,6,7,8), cex=c(0.2,1,2,4))
title(main="Sumatra (Nias) Earthquake - 28 March 2005")

magnitude.time(a)
title(main="Sumatra (Nias) Earthquake - 28 March 2005")
```

Concatenation of Catalogues

Make two catalogues and concatenate into one:

```
library(sslib)
library(ssPDE)
library(ssNZ)

a <- subset.rect(NZ, minlong=165, maxlong=180, minlat=-50, maxlat=-30,
           minmag=4, minday=julian(1,1,2005), maxday=julian(6,1,2005))

b <- subset.circle(PDE, centrelong=180, centrelat=-20, maxradius=1000,
           minmag=4, minday=julian(1,1,2005), maxday=julian(6,1,2005))

as.catalogue(a, catname="temp1")
as.catalogue(b, catname="temp2")
c(temp1, temp2, catname="newcatalogue")

epicentres(newcatalogue)
```

Point Process Model

Software requires that the process is indexed by time t

There are n events occurring in the interval $(0, T)$ at times

$$0 < t_1 < t_2 < \cdots < t_n < T$$

The conditional intensity of the *ground process* is:

$$\lambda_g(t|\mathcal{H}_t) = \lim_{\delta \to 0} \frac{1}{\delta} \Pr\{N_\delta(t) > 0 | \mathcal{H}_t\}$$

where $\mathcal{H}_t$ is the process history before t and $N_\delta(t)$ is the number of events in $[t, t + \delta)$

Includes: simple Poisson, ETAS, SRM and linked SRM

Currently the "marks" are independent of $\mathcal{H}_t$

Need to extend to

$$\lambda(t, x|\mathcal{H}_t) = \lambda_g(t|\mathcal{H}_t) f(x|\mathcal{H}_t)$$

where $f(x|\mathcal{H}_t)$ is the density of the *mark* distribution

The marks could be event magnitudes, spatial location, etc

Point Process - Estimation

The log-likelihood function is

$$\log L = \sum_{i=1}^{n} \log \lambda_g(t_i|\mathcal{H}_{t_i}) - \int_0^T \lambda_g(t|\mathcal{H}_t)dt$$

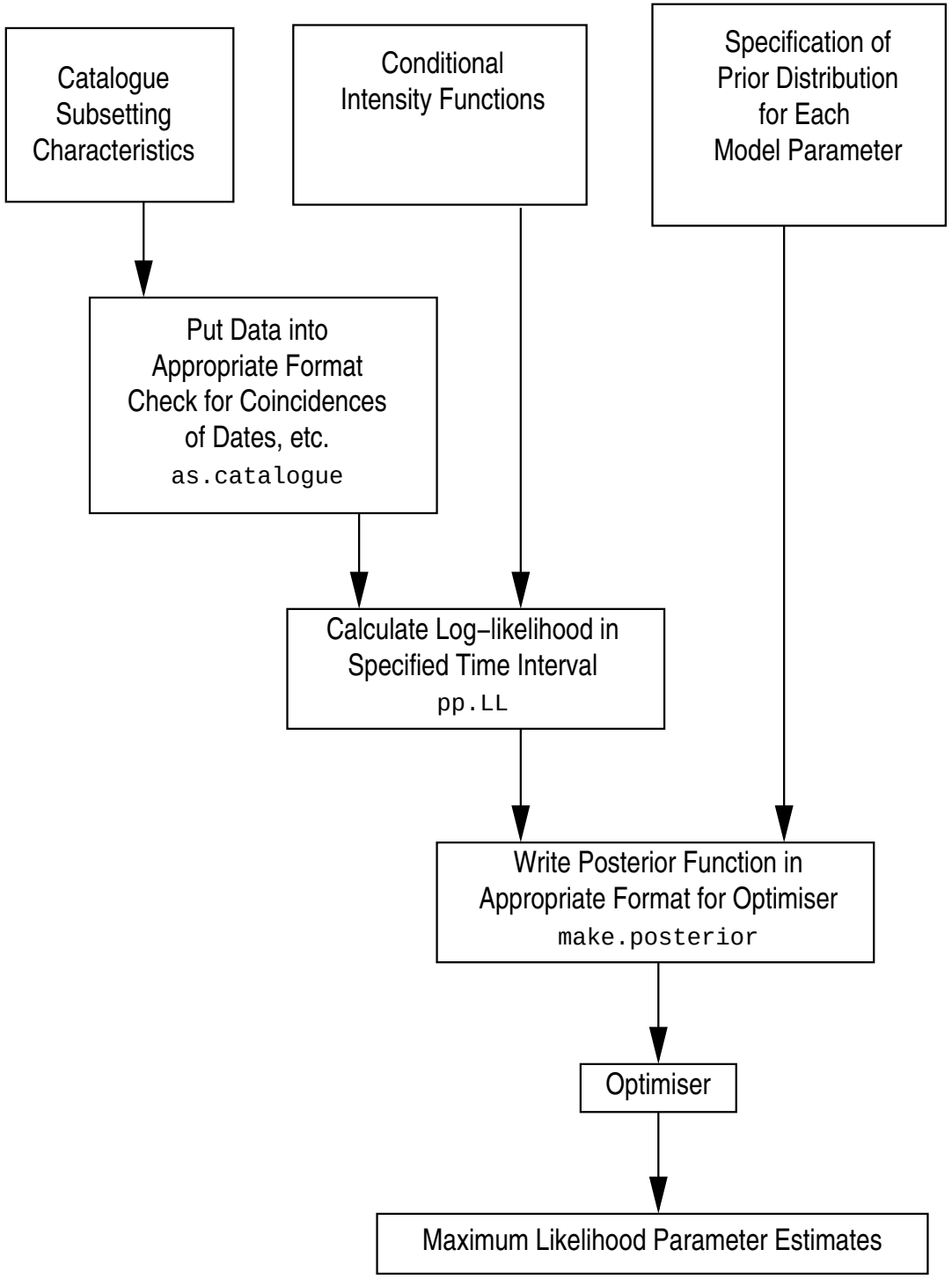
Usually solved numerically

Possible Problems:

- hard boundaries on parameter space
- little or no sample information for certain parameters
- multiple maxima
- finding satisfactory starting values

Point Process Package

Maximum Likelihood Parameter Estimation



Tangshan Event - 28 July 1976

```
library(PtProcess)
data(Tangshan)

posterior <- make.posterior(Tangshan, etas.cif, c(0, 4018))

neg.posterior <- function(params){
    x <- -posterior(params)
    if (is.infinite(x) | is.na(x)) return(1e15)
    else return(x)
}

#    initially use a grid search method
p0 <- c(0.1, 2, 1, 0.01, 1)
z0 <- optim(p0, neg.posterior, control=list(trace=1, maxit=1000))

#    steepest ascent method
z <- nlm(neg.posterior, z0$par, hessian=TRUE, typsize=abs(z0$par),
         iterlim=1000, print.level=2, gradtol=1e-8, steptol=1e-8)
```

```
x11(width=12, height=10)
par(mfrow=c(2,1), mar=c(4.1,4.1,0.5,1), oma=c(0,0,3,0))

p <- z$estimate
times <- seq(0, 4018, 1)

plot(times, log(etas.cif(Tangshan, times, params=p)), type="l",
     ylab=expression(paste(log, " ", lambda(t))),
     xlab="", xlim=c(0, 4018))

plot(Tangshan$time, Tangshan$magnitude+4, type="h",
     xlim=c(0, 4018),
     xlab="Days Since 1 January 1974", ylab="Magnitude")

title(main="Tangshan Event - 28 July 1976", outer=TRUE)
```

Point Process - Residual Process

The times of the *residual* point process are

$$\tau_i = \int_0^{t_i} \lambda_g(t|\mathcal{H}_t)dt$$

```
tau <- pp.resid(Tangshan, params=p, etas.cif)

plot(seq(1, nrow(Tangshan)), tau, type="l",
     xlab="Event Number", ylab="Transformed Time",
     xlim=c(0, nrow(Tangshan)), ylim=c(0, nrow(Tangshan)))
abline(a=0, b=1, lty=2, col="red")
```

Point Process - Simulation

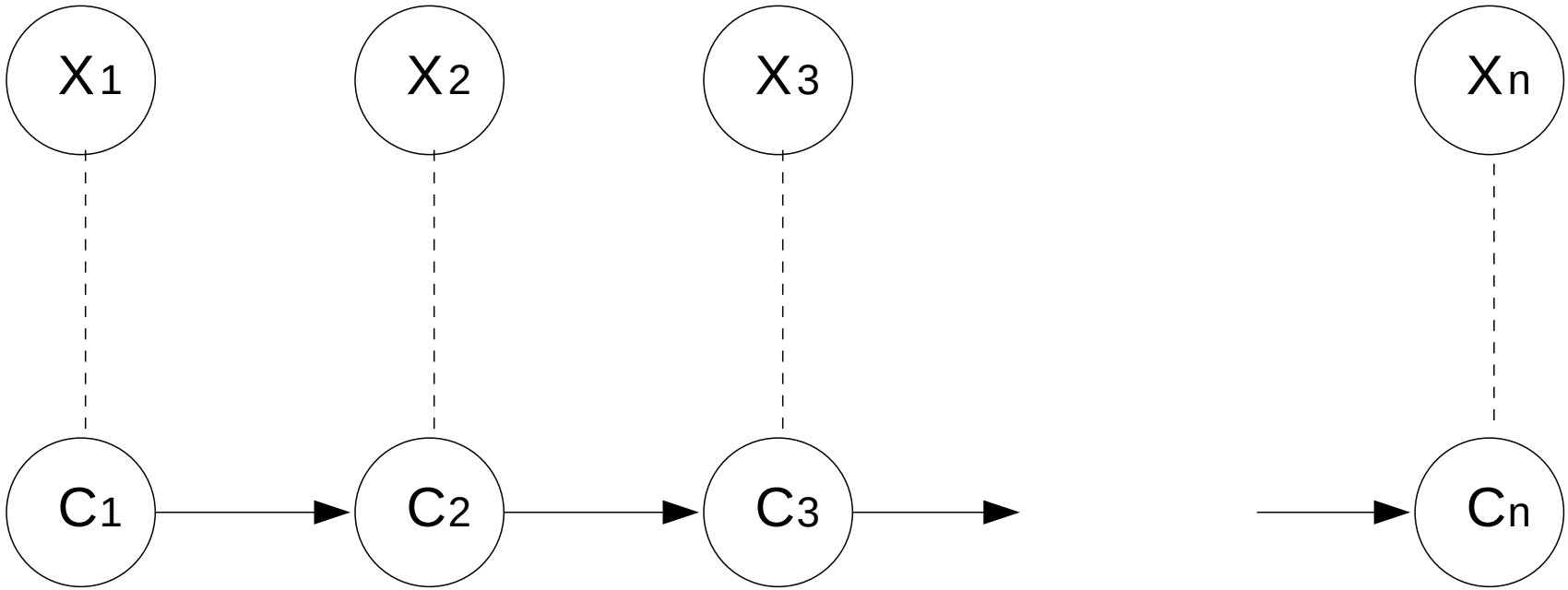
Simulate process to first event with $M \geq 5.5$ after 1985-01-01 (day 4018)

```
cond <- function(data){
    #  Stopping condition function:
    #  Must return a logical value
    return(data$magnitude[nrow(data)] >= 5.5-4)
}

x <- pp.sim(Tangshan, p, etas.cif, TT=c(4018, Inf), seed=5,
            magn.sim=1, stopping.condition=cond)

#    number of days after 1985-01-01 (day 4018) is
print(x$time[nrow(x)]-4018)
```

Hidden Markov Models

Observed sequence $\{X_i\}$ has probability distribution that is dependent on the current state of the hidden Markov chain $\{C_i\}$, m states

Distributions for X_i that can currently be handled are: beta, binomial, exponential, gamma, log normal, logistic, Gaussian, and Poisson



HMM Estimation Using EM Algorithm

Treat $\{C_i\}$ like "missing" data

Likelihood Function:

$$\begin{aligned} L &= \Pr\left\{X^{(n)} = x^{(n)}\right\} \\ &= \Pr\{X_1 = x_1, \cdots, X_n = x_n\} \\ &= \delta^{(1)} D_1 (\Gamma D_2)(\Gamma D_3) \cdots (\Gamma D_n) 1' \end{aligned}$$

where

$$D_i = \text{diag}\left(\Pr\{X_i = x_i \,|\, C_i = 1\}, \cdots, \Pr\{X_i = x_i \,|\, C_i = m\}\right)$$

"Complete" Data Likelihood:

$$\begin{aligned} L_c &= \Pr\left\{X^{(n)} = x^{(n)}, C^{(n)} = c^{(n)}\right\} \\ &= \Pr\{X_1 = x_1, \cdots, X_n = x_n, C_1 = c_1, \cdots, C_n = c_n\} \\ &= \delta^{(1)}_{c_1} \gamma_{c_1 c_2} \gamma_{c_2 c_3} \cdots \gamma_{c_{n-1} c_n} \prod_{i=1}^{n} \Pr\{X_i = x_i \,|\, C_i = c_i\} \end{aligned}$$

Now let

$$u_{ij} = \begin{cases} 1 & \text{if } C_i = j \\ 0 & \text{otherwise} \end{cases}$$

$$v_{ijk} = \begin{cases} 1 & \text{if } C_{i-1} = j \text{ and } C_i = k \\ 0 & \text{otherwise} \end{cases}$$

Then

$$\log L_c = \sum_{j=1}^{m} u_{1j} \log \delta_j^{(1)} + \sum_{j=1}^{m} \sum_{k=1}^{m} \left(\sum_{i=2}^{n} v_{ijk} \right) \log \gamma_{jk}$$

$$+ \sum_{j=1}^{m} \sum_{i=1}^{n} u_{ij} \log \Pr\{X_i = x_i \,|\, C_i = j\}$$

$$\overbrace{\Pr\left\{X^{(n)} = x^{(n)}, C^{(n)} = c^{(n)}\right\}}^{\text{maximise in M-step}}$$

$$= \underbrace{\Pr\left\{C^{(n)} = c^{(n)} \,\middle|\, X^{(n)} = x^{(n)}\right\}}_{\text{estimate in E-step}} \Pr\left\{X^{(n)} = x^{(n)}\right\}$$

E-Step:

$$\begin{aligned}\widehat{u}_{ij} &= \mathsf{E}\left[U_{ij} \,\middle|\, \widehat{\Theta}\right] \\ &= \Pr\left\{C_i = j \,\middle|\, X^{(n)} = x^{(n)}, \widehat{\Theta}\right\}\end{aligned}$$

$$\begin{aligned}\widehat{v}_{ijk} &= \mathsf{E}\left[V_{ijk} \,\middle|\, \widehat{\Theta}\right] \\ &= \Pr\left\{C_{i-1} = j, C_i = k \,\middle|\, X^{(n)} = x^{(n)}, \widehat{\Theta}\right\}\end{aligned}$$

M-Step:

estimate new values of $\widehat{\Theta}$ by *maximising* L_c

Numerical Example in SSLib

Two state Markov chain, transition probability matrix Π, observed exponential variables

```
Pi <- matrix(c(0.8, 0.2,
               0.3, 0.7),
             byrow=TRUE, nrow=2)

p <- c(2, 0.1)

x <- sim.hmm(1000, 2, Pi, "exp", list(rate=p))

#    use above parameter values as initial values
y <- Baum.Welch(x$x, Pi, c(0, 1), "exp",
                list(rate=p), maxiter=100)
```

M-Step in SSLib

All functions are generic except for the M-step

```
Mstep.exp
function (x, cond, pm, pn)
{
    rate <- apply(cond$u, MARGIN = 2, FUN = sum)/
                    as.numeric(matrix(x, nrow = 1) %*% cond$u)
    return(list(rate = rate))
}
```

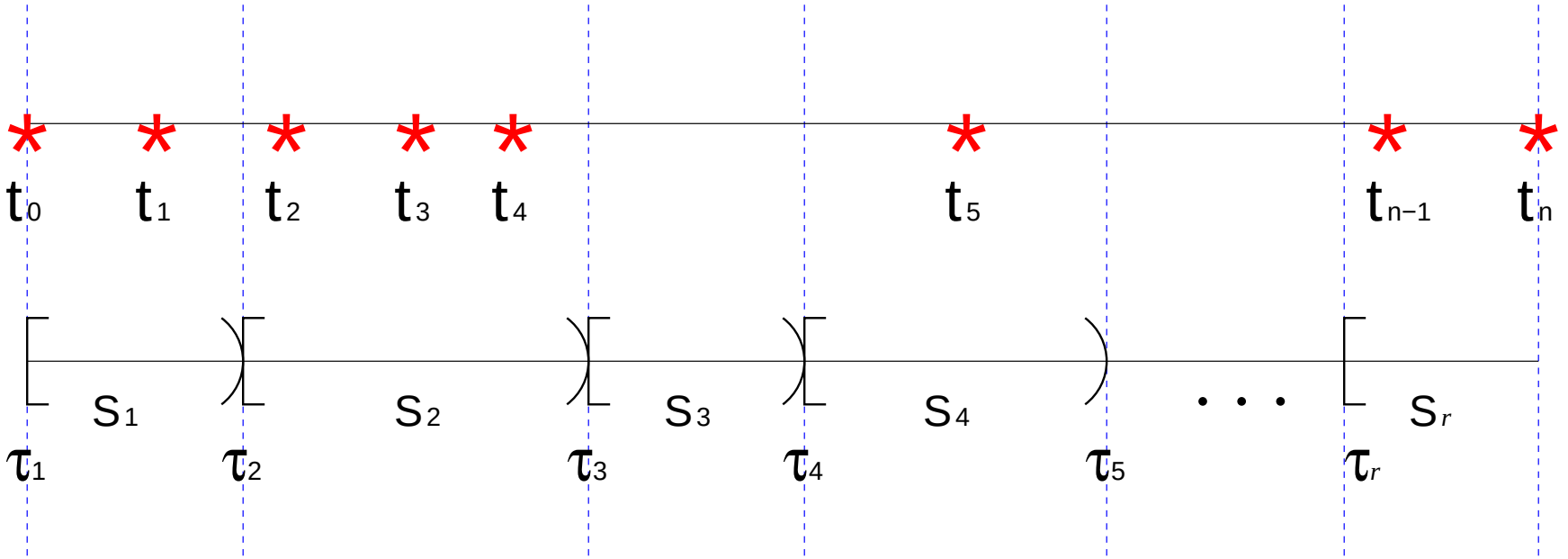
```
Mstep.pois <- function(x, cond, pm, pn){
    lambda <- as.numeric(matrix(x, nrow=1) %*% cond$u)/
                    apply(cond$u, MARGIN=2, FUN=sum)
    return(list(lambda=lambda))
}
```

Currently contains M-step functions for: Beta, Binomial, Exponential, Gamma, Log Normal, Logistic, Normal and Poisson distributions

Markov Modulated Poisson Process (MMPP)

Have an unobserved (hidden) continuous time Markov process with discrete states

Observed event times generated by a simple Poisson process (red stars) where the rate parameter is determined by the current state of the Markov process



Generator matrix Q

Has elements $q_{jk} \geq 0$ for $j \neq k$

Diagonal elements satisfy

$$q_{jj} = -\sum_{\substack{k=1 \\ k \neq j}}^{m} q_{jk}$$

$-q_{jj}$ is the (exponential) rate that the process leaves state j

$-q_{jk}/q_{jj}$ is the transition probability from state j to k

Poisson event rates for each Markov state: $\Lambda = \text{diag}(\lambda_1, \cdots, \lambda_m)$

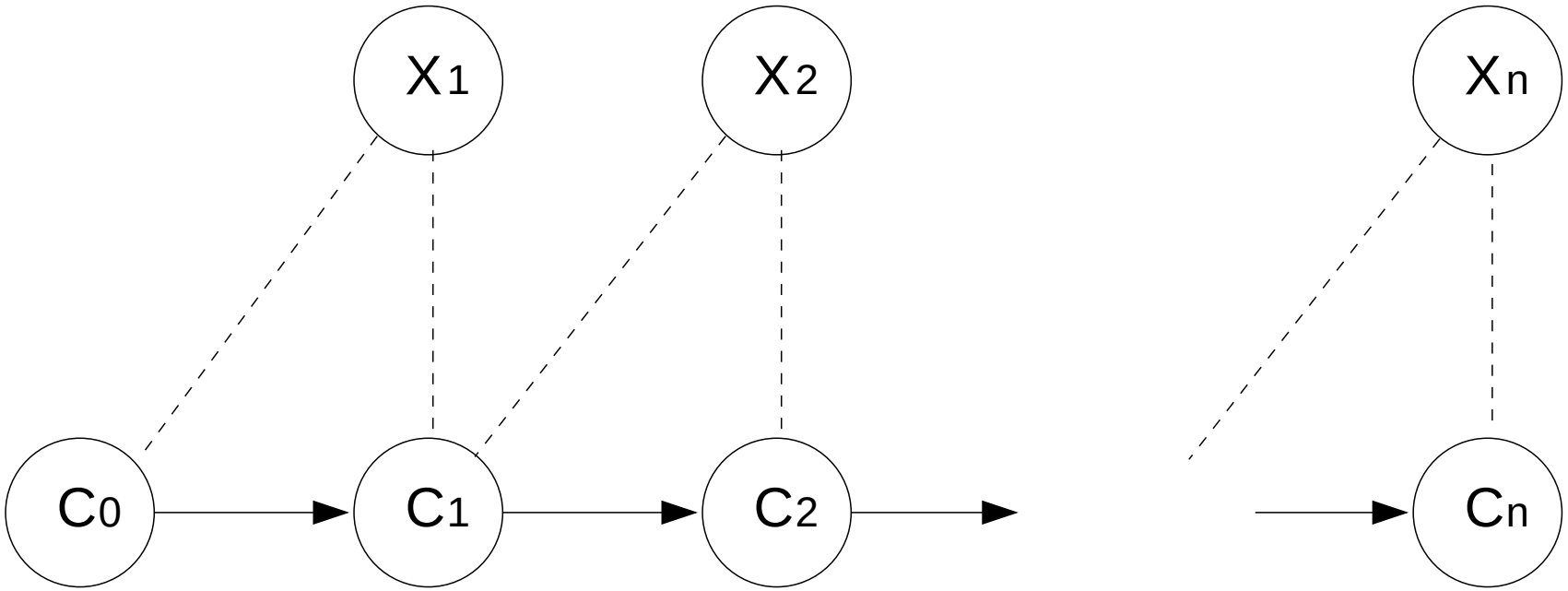
MMPP Example in SSLib

Q is the infinitesimal generator matrix of the Markov process, Λ contains the rates of the Poisson processes and δ is the marginal distribution of the Markov states

```
Q <- matrix(c(-2,  2,
               1, -1),
            byrow=TRUE, nrow=2)/10

m <- ncol(Q)
lambda <- c(5, 1)
delta <- c(0.5, 0.5)
n <- 1000

#   simulate a MMPP process
set.seed(5)
x <- sim.mmpp(n, 2, Q, lambda)

tau <- x$tau[-1] - x$tau[-(n+1)]
y <- Baum.Welch.mmpp(tau, Q, delta, lambda, prt=TRUE,
                     maxiter=100, tol=1e-8)
```

Markov Renewal Process

The MMPP is a *Markov renewal process* where the interevent times are only dependent on C_i and C_{i-1} and where $C_i = S(t_i)$



Software currently fits discrete time models and the MMPP

Extensions

Work in progress:

1. Use more general renewal structure

2. Inclusion of "marks" with a distribution in the exponential family (includes normal, gamma, beta, Poisson, binomial)

References

1. Brownrigg, R. & Harte, D.S. (2005). Using R for statistical seismology. *R News* **5(1)**, 31–35.

2. Harte, D.S. (2005). *Package PtProcess: Time Dependent Point Process Modelling. Manual of Function Documentation.* Statistics Research Associates, Wellington.
   http://homepages.paradise.net.nz/david.harte/SSLib

3. Harte, D.S. (2006). *Package HiddenMarkov: Hidden Markov Models. Manual of Function Documentation.* Statistics Research Associates, Wellington.
   http://homepages.paradise.net.nz/david.harte/SSLib

4. R Development Core Team. (2003). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna. ISBN 3-900051-07-0. URL: http://www.r-project.org.